

KNOWLEDGE-BASED ASSISTANCE IN COSTING THE SPACE STATION  
DMS

N91-10611

Troy Henson and Kyle Rone

IBM Corporation  
3700 Bay Area Blvd  
Houston, TX 77058

158

14196296

ABSTRACT

The Software Cost Engineering (SCE) methodology developed over the last two decades at IBM Systems Integration Division (SID) in Houston is utilized to cost the NASA Space Station Data Management System (DMS). An ongoing project to capture this methodology, which is built on a foundation of experiences and "lessons learned", has resulted in the development of an internal-use-only, PC-based prototype that integrates algorithmic tools with knowledge-based decision support assistants. This prototype SCEAT (Software Cost Engineering Automation Tool) is being employed to assist in the DMS costing exercises. At the same time, DMS costing serves as a forcing function and provides a platform for the continuing, iterative development, calibration, and validation and verification of SCEAT. The data that forms the cost engineering database is derived from more than 15 years of development of NASA Space Shuttle software, ranging from low criticality, low complexity support tools to highly complex and highly critical onboard software.

INTRODUCTION

Software cost engineering (SCE) is the systematic approach to the estimation, measurement, and control of software costs on a project. This discipline provides the vital link between the concepts of economic analysis and the methodology of software engineering. The tasks involved in software cost engineering are complex, and individuals with the knowledge and skill required are scarce (1). The accuracy and consistency of the SCE results are often questionable (2). There is a definite need for tools to enable SCE by managers and planners who are not experts and to improve the results (3).

PROBLEM DESCRIPTION

Software costing is required for the Space Station Data Management System, as in other projects, in many situations. Often the costing is needed within a limited time frame for a proposal, to build a business case, or to evaluate a project that is in trouble or potentially may have a problem meeting cost and schedule constraints if not adjusted. Quantitative es-

timates are required; however, little solid information may be available. A detailed analysis of the software requirements may take weeks if not months. Also, there may be a genuine concern about how well the software requirements are defined and how stable are those requirements.

To further complicate the situation the estimation process itself carries some inherent risks. Some of the factors that increase risk are software size, complexity and criticality.

Software size, particularly in a system such as the Space Station DMS, is an important factor that can ultimately affect the accuracy of the cost estimate. As the project size increases the interdependency among various elements of the software increases. Problem decomposition, an important step in the costing process, becomes more difficult.

Complexity, i.e., the relative difficulty of the software application, is an important factor affecting development costs. Some types of software are inherently more difficult to develop than others, e.g., development of an operating system compared to the development of utility software. The type of software function, such as real-time, input/output, batch, or computational, and the level of difficulty of the requirements also significantly influence software complexity.

The criticality of the software directly affects the cost of validation and verification as well as indirect costs. Software for certain medical diagnosis or treatment systems, for air traffic control, or for the Space Shuttle Flight Control System must not fail or human lives will be lost. In contrast, an inventory control system should not fail, but the impact of the failure would not result in the loss of human life.

Viable software costing depends on a quantitative historical database. If no historical data exists, the cost estimation rests on a very shaky foundation. For Space Station DMS, as for other IBM SID Houston projects, the cost engineering database is based on more than 15 years of development of NASA Space Shuttle software, ranging from low criticality, low complexity support tools to highly complex and highly critical onboard software (4), (5).

## KNOWLEDGE-BASED SCE AUTOMATION -- SCEAT DEVELOPMENT

Currently at IBM SID in Houston, software cost engineering tasks are performed by a domain expert using his/her experience and data compiled from previous efforts. For a software costing exercise, the domain expert may use stored data and algorithmic/model-based, costing programs; but a significant part of the process is based on non-automated expertise. Software costing expertise is needed in many situations, and the costing is often needed within a limited time frame. Yet, individuals with the knowledge and skill to conduct a software costing exercise are scarce. The knowledge-based decision support assistants in SCEAT identify and preserve the domain experts' knowledge, assist managers and planners who are not costing experts, and improve the accuracy and consistency of the cost estimation results.

As part of the knowledge acquisition process, the first draft of a software cost engineering workbook has been written and utilized as high-level requirements for SCEAT. The overall SCE process was analyzed from a modular/structural/dependencies viewpoint. Included is the relationship of SCE methodologies to other parts of software/systems engineering process control, at one end of the spectrum, and the decomposition of SCE into component tasks and the identification of the SCE foundation or central core, at the other end of the spectrum (See Figures 1 and 2). Then a concise approach to software cost estimation, which covers the total costs -- direct and indirect -- over the complete life cycle, using existing methodologies and tools and quantification of the primary domain expert's knowledge (6), (7) was defined. The experience-based tasks in the SCE process were identified, and the functional design of SCEAT includes expert systems to assist in those tasks. The core development cost estimation methodology was defined in the SCE workbook in more detail and implemented in the initial SCEAT prototype, which includes prototypes of expert systems for assistance in determining software criticality and software complexity.

The SCEAT prototype integrates, under Professional Work Manager (PWM) and EZ-VU on a PC, algorithmic SCE tools with expert systems for decision support assistance. SCEAT integrates the decision support assistant expert systems for software criticality and complexity determination and "stubs" for four additional planned expert systems with nine algorithmic tools including the Matrix Method tool implemented in Lotus 1-2-3. The user interface is via panels offering cook book steps to proceed through the SCE task, selectable information and tools, help screens, and pop-up screens.

## **COSTING THE SPACE STATION DMS UTILIZING SCEAT**

The SCEAT prototype has been utilized to assist in the costing of the Space Station Data Management System (DMS), a complex software system involving a distributed environment with multiple languages and applications (8), (9). The DMS for Space Station is also affected by the requirements for long lifetime, permanent operations, remote integration, and phased technology insertion of productivity tools, applications, expert systems, etc. Major cost drivers include the large size and diversity of the software, complexity, development support environment, off-the-shelf and reusable software, and criticality, which varies from one module to another. An example of the type of results -- at the end of the intermediate step of development cost estimation -- obtained with SCEAT for the DMS costing is included in the presentation.

## **SUMMARY/CONCLUSIONS**

The software cost engineering methodology employed by the domain experts at IBM SID Houston has been captured and integrated into a prototype tool SCEAT (Software Cost Engineering Automation Tool). This PC-based tool integrates algorithmic tools with expert systems which serve as decision support assistants.

SCEAT has been employed to assist in the costing of the Space Station DMS (Data Management System). It is providing a standardized approach for the DMS costing, which involves several individuals. It has made the costing process more efficient and has relieved the demands on the principal domain expert's time, allowing him to move forward into other areas of software/systems engineering process control improvement. The automation and captured methodology domain knowledge has established the foundation and mechanism enabling the continuing calibration and improvement in accuracy and consistency for Space Station DMS costing.

Plans for the future include developing additional knowledge-based decision support assistants and a tutorial to accompany the next version of SCEAT. The approach is also being expanded to other areas of software/systems engineering process control, starting with quality estimation, scheduling and management, and eventually extending to management of performance, product, resources, risk, planning, schedule. (See Figure 1). This is a continuation of the effort to accomplish the long range objective which is to automate, including the development and utilization of knowledge-based systems to serve as decision support assistants, software and systems engineering process control. Results will continue to be applied to assist in the costing and management of the Space Station Data Management System (DMS).

## REFERENCES

1. DeMarco, T., Controlling Software Projects, Yourdon, New York, 1982.
2. Kemerer, C. F., "An Empirical Validation of Software Cost Estimation Models," COMMUNICATIONS of the ACM, Vol. 30, No. 5, May 1987, pp 416-429.
3. Boehm, B. W., "Improving Software Productivity," COMPUTER, Vol. 20, No. 9, September 1987, pp 43-57.
4. Madden, W. A. and K. Y. Rone, "Design, Development, Integration: Space Shuttle Primary Flight Software System," COMMUNICATIONS of the ACM, Vol. 27, No. 9, September 1984, pp 914-925.
5. Spector, A. and D. Gifford, "The Space Shuttle Primary Computer System," COMMUNICATIONS of the ACM, Vol. 27, No. 9, pp 874-901.
6. Rone, K. Y., "A Cost Engineering Overview," class notes and handouts, March 29-30, 1988, IBM SID, Houston, Texas.
7. Rone, K. Y., "Software Engineering Process Control," copy of presentation, IBM FSD, Houston, Texas, 59 pages.
8. Chevers, E., "Avionic Systems Test Beds for Space Station," Plenary Presentation at the JAIPCC (Joint Applications in Instrumentation, Process, and Computer Control), sponsored by the IEEE, ISA, and the University of Houston - Clear Lake (UH-CL), March 12, 1987, UH-CL, Houston, Texas.
9. Heer, E. and H. Lum, "Raising the AIQ of the Space Station," Aerospace America, January 1987, pp 16-17.

THE VIEWGRAPH MATERIALS  
FOR THE  
T. HENSON PRESENTATION FOLLOW



# SOFTWARE/SYSTEMS ENGINEERING PROCESS CONTROL STRUCTURE

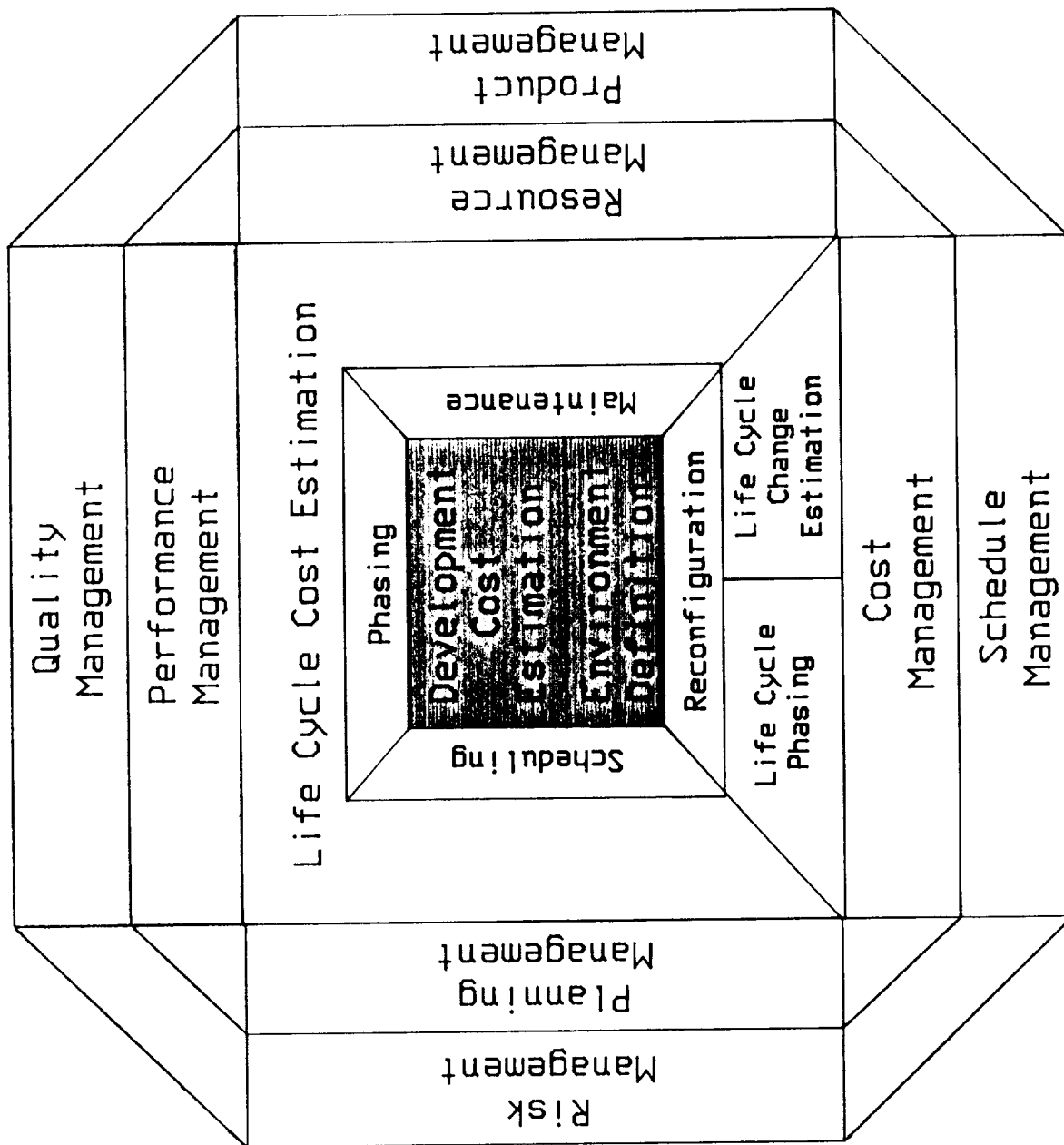


FIGURE 1.

ORIGINAL PAGE IS  
OF POOR QUALITY

T. Henson  
IBM  
7 of 19

# SOFTWARE COST ENGINEERING

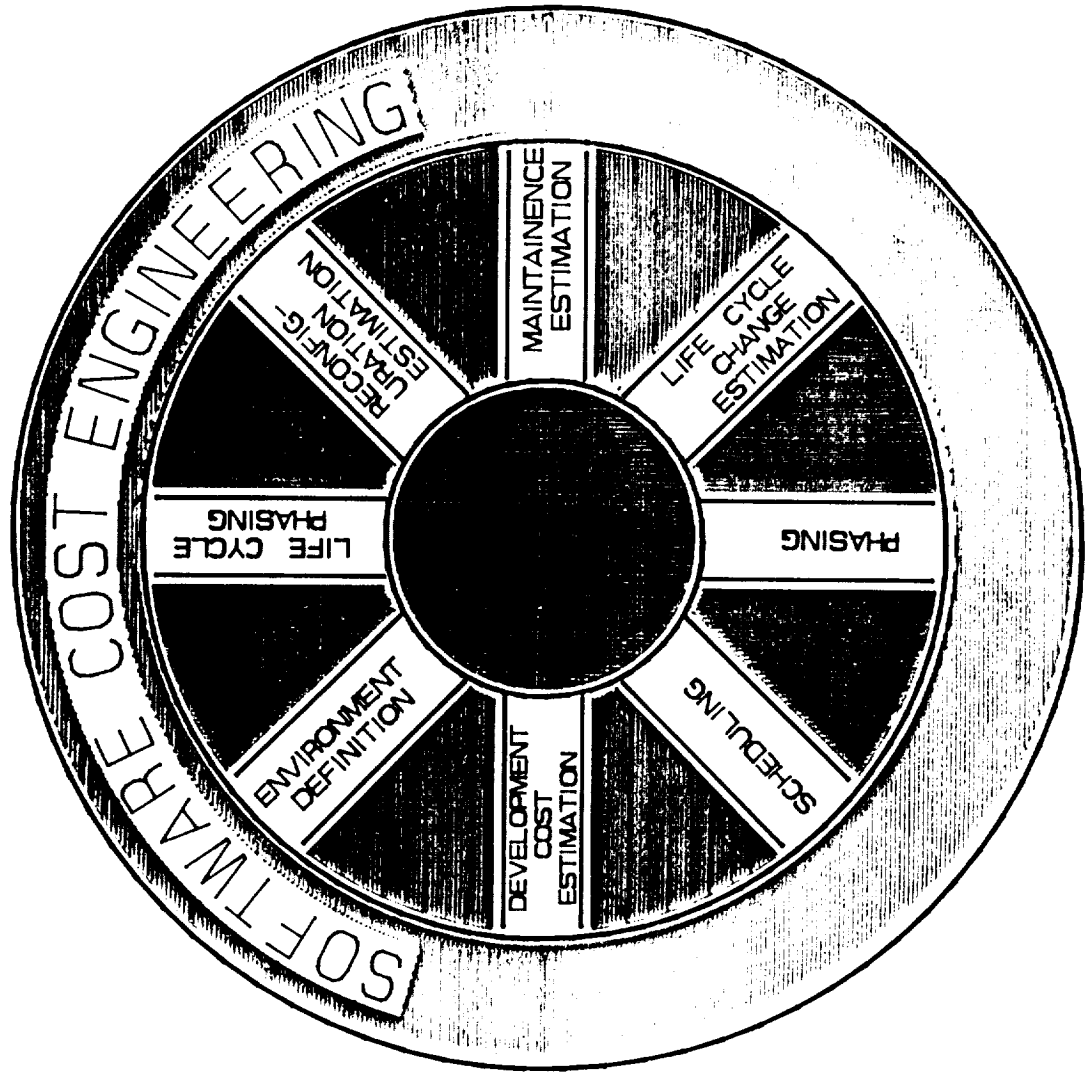


FIGURE 2.

T. Henson  
IBM  
8 of 19



**KNOWLEDGE-BASED ASSISTANCE  
in COSTING the  
SPACE STATION  
DATA MANAGEMENT SYSTEM**

**Troy Henson and Kyle Rone**

**IBM Corporation  
3700 Bay Area Blvd.  
Houston, TX 77058-1199  
PHONE: (713) 282-7476**

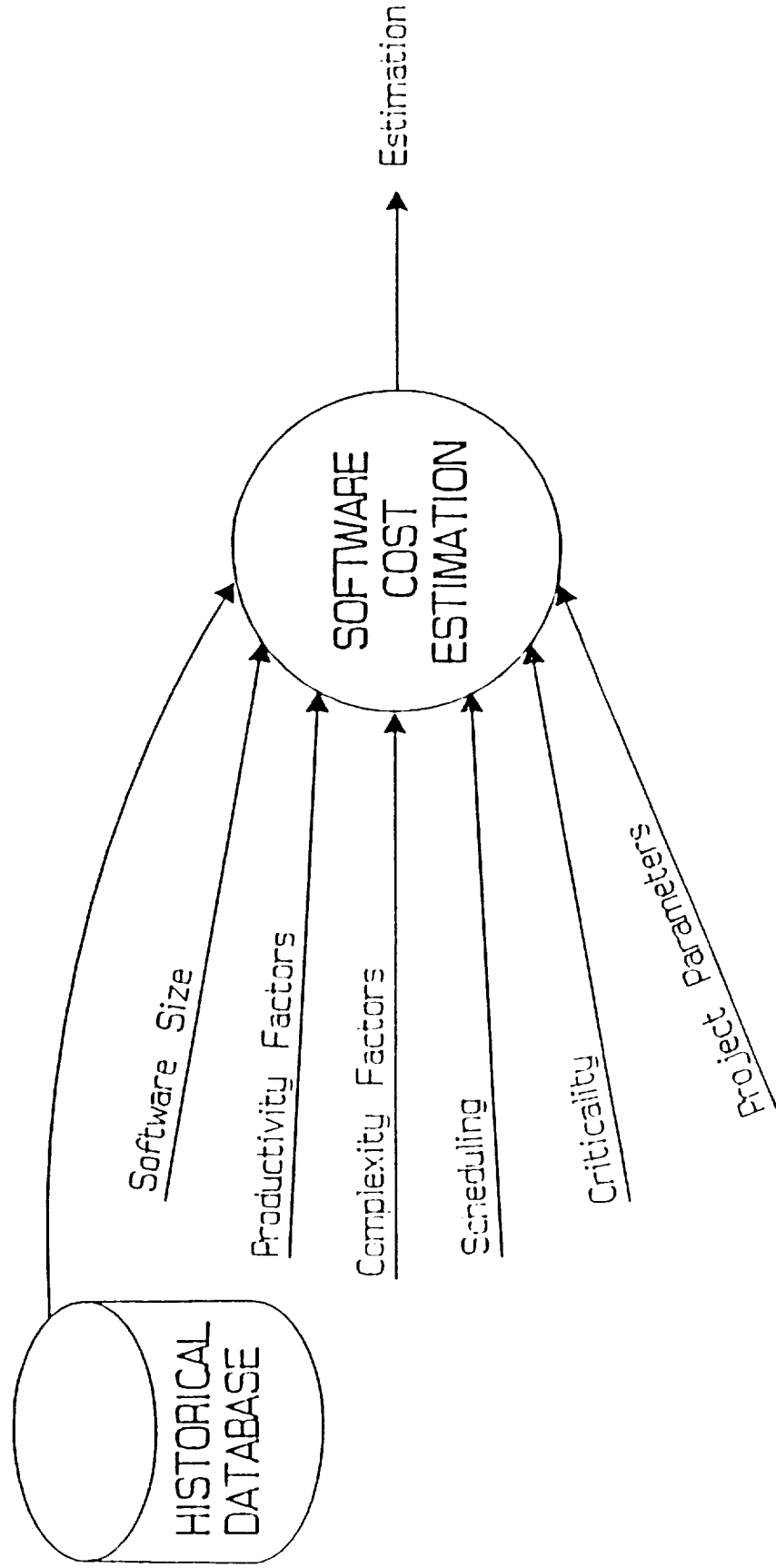
**13TH ANNUAL SOFTWARE ENGINEERING WORKSHOP**

# BACKGROUND

---

T. Henson  
IBM  
10 of 19

## What Is The Problem ?



# KNOWLEDGE-BASED SCE AUTOMATION

---

- ◆ **ACTIVITY DESCRIPTION**
  - Standardize and automate software cost engineering (SCE)
- ◆ **OBJECTIVES**
  - Define SCE process
  - Automate SCE
  - Provide SCE courses
- ◆ **CURRENT STATUS**
  - Developed SCE courses
  - Developed 9 algorithmic PC tools
  - Developed LOTUS Matrix Method tool
  - Developed criticality and complexity expert systems
  - Developed SCE workbook
  - Integrated tools and expert systems into SCE Automation Tool (SCEAT)
- ◆ **FUTURE PLANS**
  - Develop additional decision support assistants for SCEAT
  - Develop a SCEAT tutorial
  - Expand SCEAT into a process control tool

# SPACE STATION DMS DRIVERS

---

T. Henson  
IBM  
12 of 19

- ◆ Long Life
- ◆ Permanent Operations
- ◆ Remote Integration
- ◆ Distributed Environment
- ◆ Design-to-Cost System
- ◆ Multiple Levels of Criticality
- ◆ Phased Technology Insertion
  - Automation (Productivity Tools)
  - Ada
  - COTS
  - Reusability
  - Expert Systems
  - Data Base Technology

# DMS COST DRIVERS

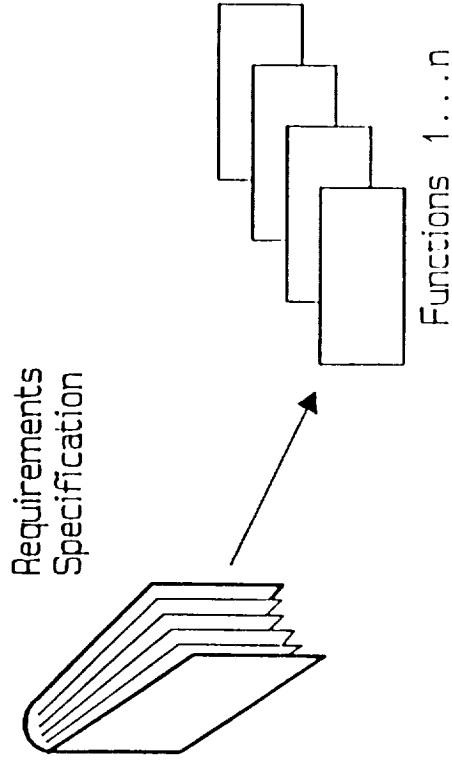
---

- ◆ Productivity Levels Driven by
  - Language
  - Complexity
  - Release (Availability of SSE Tools)
  - COTS/Reuse
- ◆ Independent Verification Driven by
  - Language
  - Complexity
  - Release (Availability of SSE Tools)
  - COTS/Reuse
  - Criticality
- ◆ Indirect Costs Driven by
  - Criticality

# COSTING METHODOLOGY

T. Henson  
IBM  
14 of 19

## Software Development Cost Estimation



For Each Function

- Estimate Size In SLOCs
- Assign Criticality Level!
- Assign Complexity Level!
- Assign Functions To Releases
- Select Productivity Factor
- Select Verification Factor
- Select Indirect Factor

Calculate Man Months

# SOFTWARE COMPLEXITY DETERMINATION ASSISTANT ES

---

- ◆ Software Complexity Based on Three Factors
  - Critical constraints
  - Interface requirements
  - Software classification
- ◆ Each Factor Assigned a Numerical Weight Range
- ◆ Each Factor Considered and Assigned a Numerical Value
- ◆ Values are Totaled to Determine the Software Complexity Rating
- ◆ Knowledge Captured in the Form of Intelligent Questions

# SOFTWARE CRITICALITY DETERMINATION ASSISTANT ES

---

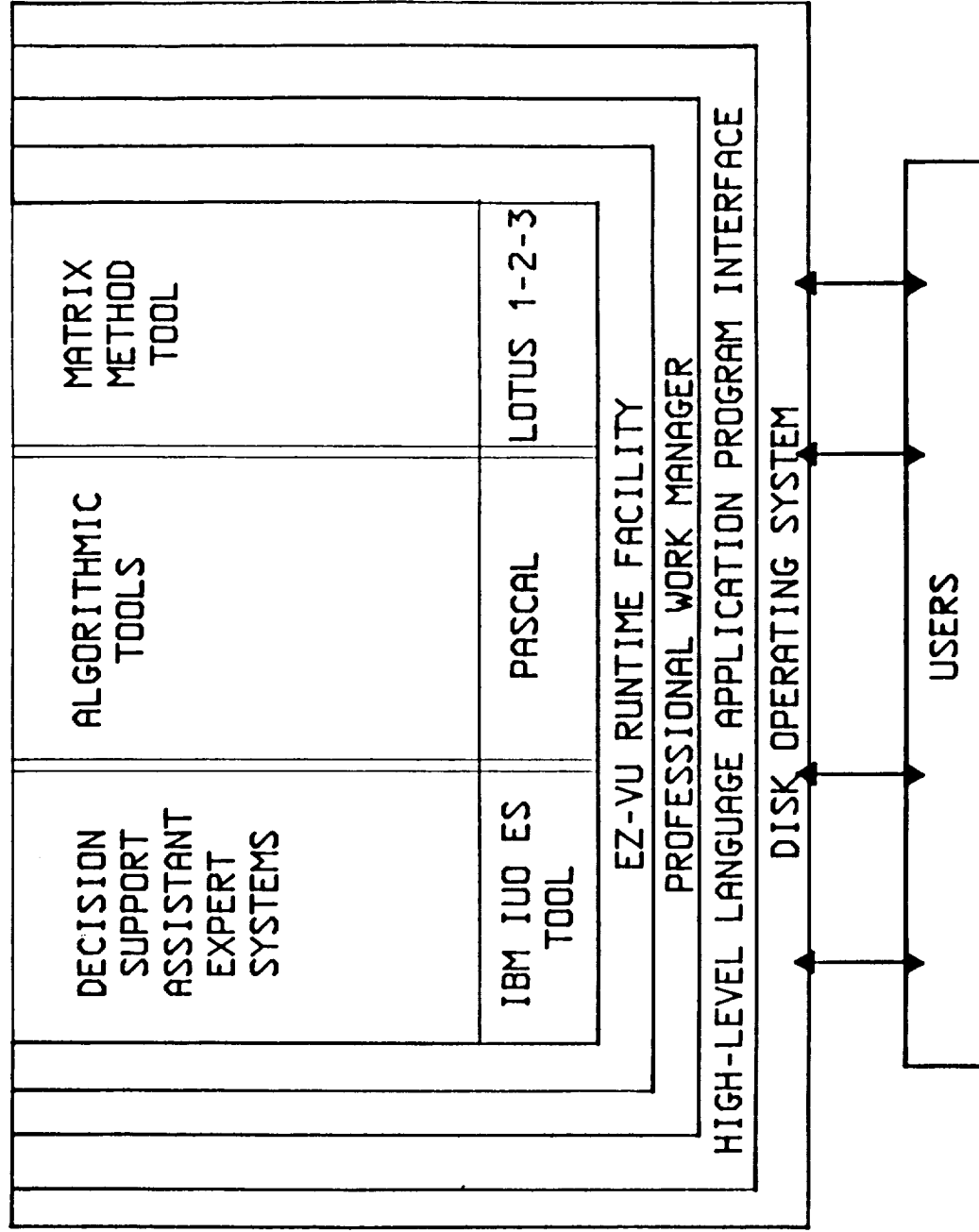
T. Henson  
IBM  
16 of 19

- ◆ **Software Criticality Based on Four Factors**
  - **Human-rated**
  - **Completion of operational objectives**
  - **Software distribution requirements**
  - **Software backup requirements**
- ◆ **Each Factor Assigned a Numerical Weight Range**
- ◆ **Each Factor Considered and Assigned a Numerical Value**
- ◆ **Values are Totaled to Determine the Software Criticality Rating**
- ◆ **Knowledge Captured in the Form of Intelligent Questions**



# SCEAT ARCHITECTURE

## SCEAT ARCHITECTURE



# EXAMPLE OF TYPE OF INTERMEDIATE RESULTS

T. Henson  
IBM  
18 of 19

## S U M M A R Y C H A R T

AREA	NEW	COTS	LANG.	COMP.	CRIT.	REL.	DEV.	VERIF.	INDIRECT	TOTAL
NOS	59600		ADA	C	SC	1,2,3	202.9	162.2	548.0	913.1
RTOS	7600	40900	ADA	C	SC	1	77.9	62.3	210.6	350.8
Standard Services	4000		ADA	L	M	1,3	10.9	4.4	11.4	26.7
Reconfig. Services	24500		ADA	M	SC, MC, M	1,2,3	78.4	50.6	162.1	291.1
DMS SM	30000		ADA	C, M, L	MC, M, L	1,2,3	115.0	62.8	191.2	369.0
DBMS	22600	50000	ADA	M	M	1,2,3	70.1	28.0	73.6	171.7
DMA	57600		ADA	C, M, L	M	3	101.5	40.6	106.6	248.7
User Services	8000	320000	ADA	M, L	SC, L	1,2,3	121.6	23.5	77.5	222.6
Recon Data Base	100000		46L	M	L	1	210.5	21.1	69.5	301.1

ORIGINAL PAGE IS  
OF POOR QUALITY

TOTALS: 313700 410900

----- 988.8 455.5 1450.5 2394.6 -----

---

## **SUMMARY**

- ◆ **Software Cost Engineering methodology has been captured and integrated in a prototype tool SCEAT.**
  - **PC-based**
  - **Integrates algorithmic tools with expert systems which serve as decision support assistants**
- ◆ **SCEAT has been utilized in costing the Space Station Data Management System.**
  - **Standardized approach**
  - **Improved efficiency, accuracy and consistency**
  - **Captured rationale**
  - **Foundation enabling continuing calibration and improvement**
- ◆ **Will be expanded into other areas of Software/Systems Engineering Process Control.**

